ETH zürich

Networked Systems
ETH Zürich — seit 2015

Communication Networks

Prof. Laurent Vanbever

Solution: Exercise 10 – TCP and Congestion Control

10.1 TCP Warm-up (Exam Question 2019)

For the following questions answer either with true or false and give an explanation for your decision. Note that in the actual exam you only had to pick true, false or do not give an answer at all (as wrong answers result in point deductions).

a) In contrast to the GBN protocol used in the project, TCP's sequence number often increases by more than one between two consecutive data packets.

Solution: True. The TCP sequence number correlates to the current byte in the transmitted byte stream.

b) A client having an ongoing TCP connection to a server (IP 1.2.3.4, port 80) is not able to start a second TCP connection towards 1.2.3.4:80.

Solution: False. The source port of the two flows can be different to distinguish the two connections.

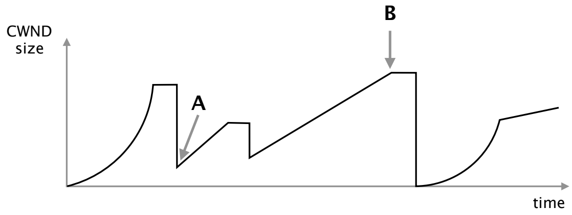
c) A TCP packet with a value of 9 in its header length field (HdrLen, sometimes also called data offset) indicates 16 bytes of TCP options.

Solution: True. HdrLen of 9 indicates 36 bytes. The default TCP header size is 20 bytes (no options) so we have 16 bytes for TCP options.

d) Consider an ongoing TCP flow. Whenever the congestion window increases, the sender can transmit additional data segments.

Solution: False. It also depends on the sender/receiver window sizes. It could e.g., be that the receiver currently cannot handle an additional data segment.

For the following four questions, we consider the congestion window (CWND) evolution observed for a flow `f` and depicted in the figure below.



e) Flow `f` was in the slow start phase exactly twice.

Solution: True. Once before point A and once after point B.

f) Flow `f` experiences at least one packet loss between time A and B.

Solution: False. The duplicated ACKs could also be due to packet reordering or delays (no packet loss required).

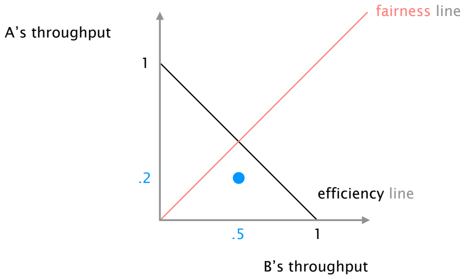
g) In the future, `f` will never be able to experience a higher CWND size than B.

Solution: False. The CWND could reach higher values in the future, for example if other ongoing flows end (more bandwidth available).

h) Consider another flow `f2` starting at exactly the same time as `f` and traversing the exact same path, then `f2` would experience the exact same CWND evolution.

Solution: False. For example the packets of the second flow could be dropped/delayed at different points in time which would result in a different CWND evolution.

10.2 Fairness



Are you getting a fair share?

Consider the situation in which two hosts, A and B, are concurrently using a 1 Mbps link with a Maximum Segment Size (MSS) of 100 kb.

Assuming that B starts with 500 kbps and A with 200 kbps (see left picture). Describe the evolution of the throughput of the two hosts when:

a) A and B rely on Additive Increase Multiplicate Decrease (AIMD).

Solution: By drawing on the left picture, you can show that AIMD eventually converges to the fairness state. For instance, assuming that both sender increase their CWND by 1 MSS when there is no congestion and divide it by 2 upon congestion, the following points can be drawn:

- (.2, .5)
- (.3, .6)
- (.4, .7) > congestion!
- (.2, .35)
- (.3, .45)
- (.4, .55)
- (.5, .65) > congestion!
- (.25, .325)
- ...

It can be seen that, because of its bigger share, B loses more than A because of the halving, eventually the system evolves along the fairness line.

b) A and B rely on Multiplicative Increase Additive Decrease (MIAD).

Solution: Again, by drawing on the left picture, you can show that MIAD does not converge to the fairness state at all, but rather to one in which A is completely shut down. For instance, assuming that both sender double their CWND when there is no congestion and decrease it by 1 MSS upon congestion, the following points can be drawn:

- (.2, .5)
- (.4, 1) > congestion!
- (.3, .9) > congestion!
- (.2, .8)
- (.4, 1.6) > congestion!
- (.3, 1.5) > congestion!
- (.2, 1.4) > congestion!
- (.1, 1.3) > congestion!
- (0, 1.2) > congestion!
- (0, 1.1) > congestion!
- (0, 1)
- ...

It can be seen that the sender which benefits from a bigger initial share will end up using the entire link.

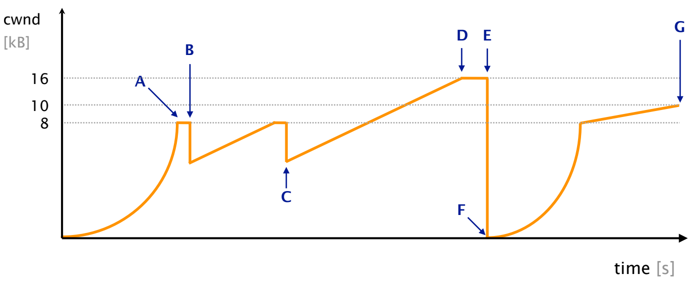
Assume now that only A is malicious, and wants to cheat congestion control to get more throughput. Describe two distinct ways A could do so and what would be the net effect on B's throughput.

Solution: There are many ways A could cheat congestion control. Two simple ones would be: *i)* increase its congestion window faster than what is prescribed, e.g. instead of increasing its CWND by 1 per RTT, it could increase it by 2 or 3 per RTT; *ii)* opening up many connections and therefore profit from more overall throughput as TCP tends to allocate throughput equally across all connections.

It is worth noting that detecting cheating is hard as controlling all end-points in the Internet is rather hopeless. Yet, the Internet continues to work and we haven't faced a new congestion collapse event since the 80s.

10.3 Congestion Window

Consider the following plot which depicts the evolution of the size of the TCP congestion window of the sender.



What kind of network conditions is this flow seeing?

Describe briefly:

a) What happens at point B?
Solution: Triple duplicate ACKs.

b) Does the event happening at point B require the network to discard packets? Why or why not?
Solution: No. This could be caused by packet reordering (e.g., due to queuing or asymmetric paths).

c) What happens at point E?
Solution: A timeout event which causes the sender to decrease its window.

d) Does the event happening at point E require the network to discard packets? Why or why not?
Solution: No. Congestion (queuing delay) in the forward or backward direction could cause the round-trip time to be higher than the retransmission timeout.

Consider that the Maximum Segment Size (MSS) of the connection is 1 kB and the Round-Trip Time (RTT) between the two end points is 100 milliseconds. The sender opens the connection at time $t = 0$. Transmission delay in this network is negligible, so you should only consider the propagation delay in the following.

e) How much time has elapsed at point A?

Solution: Total time: 1 RTT for the TCP handshake + 3 RTT in slow-start ($1 \rightarrow 2 \rightarrow 4 \rightarrow 8$ MSS) = 4 RTT, i.e. 400 ms.

f) How much time has elapsed *between* point C and D?

Solution: The congestion window grows from 4 MSS to 16 MSS linearly at a growth of 1 MSS per RTT, translating to 12 periods of RTT or 1.2 s.

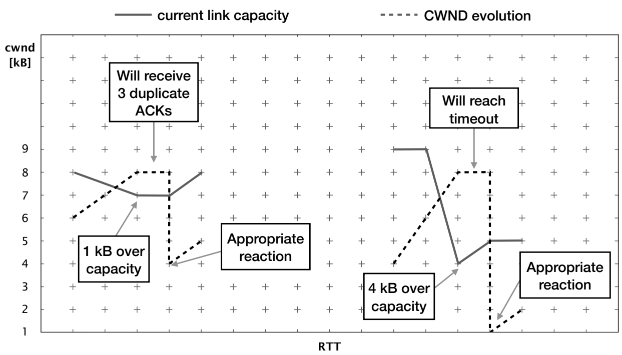
g) How much time has elapsed *between* point F and point G?

Solution: From F, we start with slow start phase to an 8K window size which takes 3 RTT ($1 \rightarrow 2 \rightarrow 4 \rightarrow 8$ MSS). After that we flip to congestion avoidance and follow an additive increase from a 8 to a 10 MSS window size (8, 9, 10 MSS) which takes 2 RTTs. In total 5 RTTs, i.e. 500 ms, elapsed from point F to G.

Briefly explain how come point D is higher than point B. Would you expect this to happen often?

Solution: D's height can vary as a consequence of other concurrent flows being sent along the same link. This is therefore likely to happen all the time.

10.4 Drawing practice (Exam Question 2018)



Reaction of the CWND (dashed line) if the current link capacity (continuous line) is exceed by at most 2 kB (left) or by more than 2 kB (right).

In this task, you will draw the Congestion Window (CWND) evolution in reaction to the available capacity of a link in a network. The CWND follows the well-known TCP congestion control algorithm using slow-start. Whenever the CWND value exceeds the current link capacity, the CWND algorithm will react in the following way:

1. The current CWND value is kept for the entire next RTT (no increase or decrease);

2a. If the current link capacity was exceeded by at most 2 kBs, the CWND algorithm will observe three duplicate ACKs during the next RTT and will react appropriately (see the figure above on the left);

2b. If the current link capacity was exceeded by more than 2 kBs, the CWND algorithm will reach its timeout during the next RTT and will react appropriately (see the figure above on the right right).

Draw the CWND evolution directly into the figure below in reaction to the link capacity indicated with the continuous line. Start at the bottom left corner (RTT 1, CWND 1 kB) and assume that the CWND corresponds to a flow that just started, e.g. you are in the slow-start phase. You can stop once you reach RTT 22. To help you, a correct portion of the CWND is plotted between RTT 11 and 14 (dashed line).

Solution: